

# The University of Nottingham

DEPARTMENT OF MECHANICAL, MATERIALS & MANUFACTURING ENGINEERING

A LEVEL 1 MODULE, AUTUMN SEMESTER 2015-2016

## **COMPUTER PROGRAMMING**

Time allowed ONE and half Hour

---

*Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced.*

### ***Answer ALL Questions***

*Only silent, self contained calculators with a Single-Line Display, or Dual-Line Display are permitted in this examination.*

*Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.*

*No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.*

***DO NOT turn examination paper over until instructed to do so***

**ADDITIONAL MATERIAL:**     None

### **INFORMATION FOR INVIGILATORS:**

Question papers should be collected in at the end of the exam – do not allow candidates to take copies from the exam room.

*Turn Over*

**SECTION 1: Command-line instructions**

For questions 1-5, state what MATLAB commands you would use to carry out the requested operations. For question 6 no MATLAB code is required, only a couple English sentences.

1. Plot a graph of  $y=e^{2x}$  between -1 and 1 using 100 data points.
2. Evaluate the equation  $y=\sin^2 a+\cos^2 a$  at  $a=45$  degrees.
3. Prompt the user to enter his/her name and display this to the screen using a second command.
4. Play the 1D array 'a' on the computers speakers at 40000 samples per second.
5. Read the jpg file 'one.jpg' from disk into an array 'data' and display the image.
6. Explain why reading a file from a standard hard disk is much slower than accessing an array from main memory.

[60 marks in total – 10 per question]

**SECTION 2: Usage of MATLAB features**

For each of the following features within MATLAB, write a very short script demonstrating how it could be used. Then in no more than ten words explain what it does.

7. break
8. while
9. if-elseif-else-end
10. sprintf for two variables of different types.
11. fprintf
12. The ' $\sim$ =' comparison command.

[60 marks in total – 10 per question]

**SECTION 3: Understanding MATLAB code**

PLEASE NOTE: line numbers such as Q13a, Q13b, Q13c etc. are provided on the left hand side. These are not a part of the MATLAB code. You should use these in your response to indicate which line of code you are interpreting. You do not have to transcribe (i.e. copy-out) the lines of code.

For each line of code below provide an interpretation of what it does. Then in no more than 40 words describe what the programmer wanted it to do. In each example there is a single mistake in the script, write out the line of code containing the mistake along with what the line should have read.

13.

|      |   |
|------|---|
| Q13a | <code>dat=load('data.dat')</code>                   |
| Q13b | <code>a=dat(:,1)</code>                             |
| Q13c | <code>b=dat(:,2)</code>                             |
| Q13d | <code>s=size(a)</code>                              |
| Q13e | <code>s=s(1)</code>                                 |
| Q13f | <code>out=zeros(s,1)</code>                         |
| Q13g | <code>for n=1:s</code>                              |
| Q13h | <code>out(n)=(b(n+1)-b(n-1))/(a(n+1)-a(n-1))</code> |
| Q13i | <code>end</code>                                    |
| Q13j | <code>c=[a out]</code>                              |

14.

|      |   |
|------|---|
| Q14a | <code>str=input('','s')</code>                          |
| Q14b | <code>s=size(str);s=s(2)-1</code>                       |
| Q14c | <code>for i=1:s</code>                                  |
| Q14d | <code>a=floor(rand(1)*s+1);b=floor(rand(1)*s+1);</code> |
| Q14e | <code>t=str(a);</code>                                  |
| Q14f | <code>str(a)=str(b);</code>                             |
| Q14g | <code>str(b)=t;</code>                                  |
| Q14h | <code>end</code>  |
| Q14i | <code>b=sprintf('The text is %f',str);</code>           |
| Q14j | <code>disp(b)</code>                                    |

[60 marks in total – 30 per question]

*Turn Over*

**SECTION 4: Writing MATLAB code**

15. You are working for a company that produces  $1\text{m}^2$  silicon solar panels, which are installed on the roofs of houses. Each solar panel has a finite lifetime. The company produces solar panels in batches of 100 and every batch is given a unique batch ID number. All the solar panels in the first batch ever produced by the factory were given a batch ID number of 1, all solar panels produced in the second batch were given a batch ID number of 2 and so on.. Every time a solar panel fails due to old age, it is shipped back to the factory and its batch ID number is recorded along with how long it survived on a rooftop before failing. This data is stored in a file called `lifetimes.dat`, where the first column contains the batch ID number of the panel, and the second column the number of hours the panel survived before failing.
- (a) Read the file `lifetimes.dat`, into an array and plot the column containing the batch ID number against the column containing the life time. Before plotting the data, you should split the data into two columns. The graph has to be correctly labeled and have a title. [25 marks]
- (b) Write a script to process the file '`lifetimes.dat`', and calculate the average time a solar panel lasts on a roof. The script must be able to accept any length of file and the task must be done using a single for loop. [25 marks]
- (c) Using a *for* or *while* loop (not the built in min/max commands), write another script to process the file '`lifetimes.dat`' and calculate the maximum and minimum times a solar panel has lasted on a roof. Write these values to a text file called '`max_min.dat`' using the `fopen` command. [25 marks]
- (d) To understand why the solar panels fail, once they are returned to the factory, they are illuminated by a high power blue light, then an image is taken of them using a high resolution camera. This image is stored in a file called '`image.dat`' as a 2D array. In this array values of over 100 are considered areas of the solar panel damaged by the sunlight, while values of under 100 are considered still working. Each element in the array represents an area of  $2\text{cm}^2$  on the solar panel. Write a script to load the file '`image.dat`', then calculate the total area damaged due to exposure to sunlight. If the area is bigger than  $100\text{cm}^2$  write to the screen the text "Solar panel damaged by sunlight" and if the area is under  $100\text{cm}^2$  write to the screen "Not damaged by sunlight". [25 marks]

*Continued on next page*

- (e) Now returning to processing the data stored in the file 'lifetimes.dat' Write a function that accepts two arrays, one containing the list of batch ID numbers (as stored in the file) and one which contains the corresponding list of solar cell life times (as stored in the file). This function should also accept an integer called `batch_number_of_interest`. After the function has been called, it should return the average lifetime of only the solar panels with a batch number equal to `batch_number_of_interest`.

[25 marks]

- (f) Using a for loop, write a script to calculate the highest batch number of solar panels in the file. Define an array of length equal to the highest batch number present in the file 'lifetimes.dat'. Then use the function defined in part e to calculate how long on average each batch of solar panels lasted on the roof. Plot a graph of batch number against average time life time.

[25 marks]

**END**